

1

《C/C++ 学习指南》

第12.1讲：动态分配内存-malloc/free

作者：邵发 QQ群：417024631

官网：http://www.afanihao.cn/c_guide/

答疑：<http://www.afanihao.cn/kbase/>

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

2

版权所有，侵权必究

困境

（回础10.4讲）

需要用一个数组来保存用户的输入，但是却不知道用户会输入多少条数据。

- (1) 如果设一个太大的数组，则显得浪费内存
- (2) 如果设得太小，又怕不够

问题：如何做到恰好够用、又一点不浪费呢？

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

困境

C/C++里，要求数组长度为常量

```
int Contact[100]; // 长度必须在代码里固定
```

不能这样：

```
int n = 0;  
scanf( "%d" , &n);  
int Contact[n]; // 编译错误！！数组长度必须为常量
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

动态内存分配 - malloc/free

系统中存在一个内存管理器(MM, Memory Manager)，它负责管理一堆闲置内存。它被设计用于解决此类问题。

MM提供的服务：应用程序可以向MM申请（借出）一块指定大小的内存，用完之后再释放（还回）。

例如，

```
void* ptr = malloc (1024); // 申请，从MM借出内存  
free(ptr); // 释放，还回MM
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

动态内存分配 - malloc/free

malloc函数

`void* malloc(int size)`

参数size: 指定要申请的内存空间的大小

返回值: `void*` , 指向这一块内存地址

(MM不关心你拿这块内存来存储何种数据, 所以返回`void*`)

应用程序在使用malloc时, 要把返回值转换成目标类型。例如, 要申请一块空间存放1000个Contact对象, 则

```
int size = 1000 * sizeof(Contact);  
Contact* p = (Contact*) malloc(size);
```

这块内存和数组没有本质区别, 用法完全相同。根据第9章, 数组本质上就是一块连续的内存, 两者是一样的。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

版权所有, 侵权必究

动态内存分配 - malloc/free

free函数

`void free(void* ptr)`

ptr: 先前malloc返回的内存地址

返回值: `void*` , 指向这一块内存地址

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

7

版权所有，侵权必究

动态内存分配 - malloc/free

举例：

先计算需要多少字节的内存空间

```
char* p = (char*)malloc(8); // 申请8个字节
for(int i=0; i<0; i++)
{
    p[i] = i + 1;
}
free(p); // 释放
```

// 观察内存：p指向的内存里的数据是显而易见的

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

8

动态内存分配 - malloc/free

举例：

```
int size = 4 * sizeof(Contact);
Contact* p = (Contact*) malloc (size);
p[0].id = 1;
strcpy(p[0].name, "shaofa");

free (p);
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

动态内存分配 - malloc/free

举例：

// 用户自己决定要输入多少条记录

```
int n = 0;
```

```
scanf( "%d" , &n);
```

// 用户需要多少，就分配多少内存

```
int size = n * sizeof(Contact);
```

```
Contact* p = (Contact*) malloc(size);
```

// 释放

```
free(p);
```

// 观察内存：p指向的内存里的数据是显而易见的

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

关于MM

(1) MM是一个系统级的东西，所有的应用程序都向同一个MM申请内存。

(2) 何为借出？实际上，在内存被借出时，MM只是把它管理的内存标记了一下，表示该段内存已经被占用。比如，它把每一段被占用的内存给记录下来（首地址，长度）

(p0, n0) (p1, n1) (p2, n2) ...

(3) MM非常慷慨：①只要有人malloc，它都同意借出 ②你不归还，它永远不会主动要求你free。

这意味着，**用户程序应该自觉得及时free**，以便不耽误别的应用程序的使用。如果有个应用程序不停地malloc，而不free，那最终会用光MM的内存。当MM没有更多闲置内存时，malloc返回NULL，表示内存已经用完。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

关于MM

MM管理的内存区域称为“堆”Heap，相当于一个仓库。当用程序要malloc时，就从仓库里登记借出。当free时，就将这一块内存放回仓库。

再次重申：应用程序在malloc之后，应该尽早free!

关于MM

为何free的时候只需一个首地址呢？为什么不传递长度？

实际上，MM对借出的内存块进行标识

(p0, n0) (p1, n1) (p2, n2) ...

它内部已经保证任意两块内存不会“交叠”，即不会重叠，不会把一块内存同时借给两个应用程序使用。

所以，每块内存的首地址都是不同的，在free的时候只需要指明首地址即可。

版权所有，侵权必究

小结

1. 使用malloc申请内存，具体申请多大的空间有你决定
2. 使用free释放内存
3. 使用原则：需要的时候再申请，不需要的时候立即释放